

Compressive Sensing tesztkörnyezet tervezése

Kalmár György

Szegedi Tudományegyetem
kalmargy@inf.u-szeged.hu

Absztrakt. A hagyományos mintavételezés szerint bármely jelet a Nyquist mintavételi törvény szerint legalább a benne jelen lévő maximális frekvencia kétszeresével kell mintavételezni ahhoz, hogy információvesztés nélkül rekonstruálható legyen. A *Compressive Sensing* (CS) ezzel szemben, a mintavételezés egy általánosabb megfogalmazásával kimondja, hogy jóval kevesebb mérés is elegendő egy adott jelen, amelyről azonban feltételezzük, hogy valamely térben ritka reprezentációval rendelkezik. A CS-hez kapcsolódó kutatások az elmúlt tíz évben jelentős figyelmet kaptak a tudomány minden területén. Az egyik legnagyobb problémát a valós körülmények közötti egyszerű tesztelhetőség hiánya okozza. Célom egy olyan tesztkörnyezet tervezése, amely a CS-gel foglalkozó kutatók számára megoldást nyújtana a valós körülmények közötti tesztelésben. A hardver megfelelő felépítése, egyszerűsége, bővíthetősége és könnyen elérhetősége elsődleges szempont. Szoftvere ehhez alkalmazkodó, modulás felépítésű, valós idejű rendszer.

1. Bevezető

A modernkori digitális fejlődés egyik fő hajtóereje a digitális jelfeldolgozás robbanásszerű fejlődése volt. A jelek érzékelése analóg digitális átalakítók (ADC) segítségével történik, amely kimenetén már számokban kifejezett értékek jelennek meg, amiket az egyre fejlődő számítógépekkel egyre gyorsabban és komplexebb módon lehet feldolgozni. Az ADC átalakításainak számát másodpercenként, vagyis a mintavételezés frekvenciáját az ún. Nyquist törvény határozza meg, amely sávkorlátos jelek esetén garantálja azok tökéletes helyreállíthatóságát a digitális mintákból. Vannak olyan alkalmazások (pl. radar), amelyekben a mintavételezési frekvencia nagyon magas és a technológia nem képes lépést tartani a növekvő igényekkel. Ezeken a területeken is megoldást nyújthat a jelérzékelésre az ún. *Compressive Sensing* (tömörítő érzékelés, CS), amely a jeleket más módon, úgynevezett lineáris mérések segítségével méri. A CS kutatások az elmúlt évtizedben kiemelkedő eredményeket hoztak. Számos alkalmazás, módszer azonban valós körülmények közt sosem tesztelt, ugyanis CS módon működő eszközök nem elérhetőek forgalomban. Szükség van tehát egy olyan általános keretrendszerre, amely valós körülményeket biztosítva, megfelelő szoftver és hardver támogatással elérhetővé teszi CS alapú tesztek, kiértékelések végrehajtását, mégpedig a CS alapulva működő eszközök emulálásával.

A következő fejezetben megismerkedünk a CS legfontosabb problémáival és eredményeivel, majd a 4. fejezetben a kialakítandó hardver, az 5. fejezetben pedig a szoftver rendszerek követelményeivel és terveivel foglalkozunk.

2. Compressive Sensing alapok

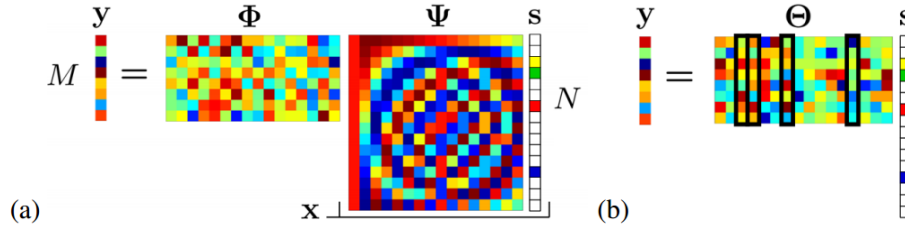
A Nyquist mintavételezési ráta felett elvégzett jelérzékelés tökéletesen leír egy jelet, kihasználva annak sávkorlátozottságát. A jelben előforduló maximális frekvencia legalább kétszeresével történő pillanatszerű mintavételezés a jelből, garantálja, hogy utána az eredeti jel triviális módon, a mintákból, információvesztés nélkül visszaállítható.[1] Ezzel szemben a *Compressive Sensing* (tömörítő érzékelés, CS) lehetőséget kínál arra, hogy jeleket jóval kevesebb mintából (mérésből) is visszaállíthassunk, kihasználva azok tömöríthetőségét.[2] A mintavételezés egy általánosabb ún. lineáris mérésre alakul. A következőkben a CS alapjaival ismerkedünk meg, bővebb információkat a [2–8] tartalmaznak.

Tekintsünk egy valós értékű, egydimenziós, véges hosszú, diszkrét idejű jelet $N \times 1$ dimenziós oszlopvektornak R^N -ben, jelöljük \mathbf{x} -szel. Elemei $\mathbf{x}[n]$, $n = 1, 2, \dots, N$. Többdimenziós jelek esetén, pl. képek, azokat vektorizáljuk egy hosszú, egydimenziós oszlopvektorra. Minden jel R^N -ben kifejezhető N darab lineárisan független $N \times 1$ -es vektor segítségével, $\{\psi_i\}_{i=1, \dots, N}$. Az egyszerűség kedvéért tekintsünk ortonormált bázist (a vektorok függetlenek, merőlegesek egymásra és egységnyi hosszúak). A $\{\psi_i\}_{i=1, \dots, N}$ vektorokat, mint oszlopvektorokat egymás mellé helyezve egy $N \times N$ -es bázismátrixot $\Psi = [\psi_1 | \dots | \psi_N]$ kapunk. Ψ használatával minden \mathbf{x} jel kifejezhető, mint

$$\mathbf{x} = \sum_{i=1}^N s_i \psi_i \text{ vagy } \mathbf{x} = \Psi \mathbf{s} \quad (1)$$

ahol \mathbf{s} $N \times 1$ -es oszlopvektor, az együtthatók vektora, $s_i = \langle \mathbf{x}, \psi_i \rangle = \psi_i^T \mathbf{x}$. Az \mathbf{s} i . eleme tehát a jel és az i . bázisvektor belső szorzatának értéke, amely bizonyos szinten azt fejezi ki, hogy mennyire hasonló a jel az adott vektorhoz. Nyilvánvaló, hogy \mathbf{x} és \mathbf{s} a jel ekvivalens reprezentációi, \mathbf{x} a jel időtartománybeli alakja, \mathbf{s} pedig a Ψ -tartománybeli alakja.

Figyelmünket olyan jelekre korlátozzuk, amelyek ritka reprezentációval rendelkeznek valamely megfelelően választott tartományban, vagyis egy \mathbf{x} jel csak K darab bázisvektor lineáris kombinációjából áll össze, és $K \ll N$. Tehát K darab elem \mathbf{s} -ben nem nulla, $(N - K)$ pedig nulla. A ritkaság valóban jellemző a tömöríthető jelekre, a természetes és ember által generált jelek legtöbbször rendelkeznek ritka reprezentációval, ahol mindössze néhány együttható vesz fel magas és befolyásoló értéket a számos elhanyagolható mellett. A tömöríthető jelek tehát jól reprezentálhatóak azért, hogy \mathbf{s} -ben csak a K legnagyobb értékű elemet őrizzük meg, a többit kinullázuk, ezt K -ritka reprezentációnak nevezzük és a napjainkban használt tömörítő eljárások (*transform coding*) alapját képezi. Például képek tömörítésére használt JPEG és JPEG-2000 szabványok rendre diszkrét koszinusz és wavelet-bázisokat használnak a képek tömör és ritka



1. ábra: (a) CS mérés véletlen Φ mátrix segítségével, Ψ a DCT bázis, s 4-ritka együtthatóvektor. (b) A mérési eljárás a Θ mátrixot használva. y a $\Theta = \Phi\Psi$ kiemelt oszlopainak lineáris kombinációja, amelyekhez tartozó s_i értékek nem zérusok.

jellemezésére, amely során csak K darab nem zérus elem értékét és helyét őrzik meg. Audiojelek és kommunikációra használt jelek esetén a Fourier-térnek hasonló szerepe van.

A transform coding napjaink jól bevált módszere a jelérzékelést és -feldolgozást tekintve. A modern jelfeldolgozás alapvetően az érzékelés, majd tömörítés (*sample-then-compress*) elvét használja, ahol a vett minták száma magas, de a jel önmagában jól tömöríthető, információs rátája alacsonyabb, mint a jelben lévő változások rátája. Ebben a szerkezetben először az x jel összes N darab mintáját összegyűjtjük, a teljes együttható vektort kiszámítjuk ($s = \Psi^T x$), megőrizzük a K legnagyobb együtthatót és eldobjuk a többi $(N - K)$ -t, kódoljuk és digitalizáljuk a K megőrzött együttható értékét és pozícióját.

Az érzékelj, majd tömöríts felfogás sajnos három alapvető hibától szenved. Az egyik, hogy hiába kicsi a megőrzendő együtthatók K darabszáma (amely jó tömörítők esetén kulcsfontosságú), akkor is a teljes N mintát össze kell gyűjteni, amely egyes alkalmazásokban hatalmas számot is jelenthet. Továbbá, a kódoló egységnek az összes s_i együtthatót ki kell számolni, pedig mindössze K darab kerül megőrzésre. Végül, az együtthatók helyének és nagyságának bonyolult kódolása lényeges erőforrás foglalással jár a kódolók számára.

A *Compressive Sensing* az említett jelérzékelés alternatívájaként egy jóval általánosabb adatgyűjtést használ, amely a jelet egyből tömörített formába gyűjti, anélkül, hogy a köztes lépések során mind az N darab minta rögzítése megtörténne. Megértéséhez tekintsünk egy sokkal általánosabb lineáris mérési eljárást, amely $M < N$ darab belső szorzatot számol ki az x jel és a $\{\phi_j\}_{j=1..M}$ vektorok közt, előállítva $y_j = \langle x, \phi_j \rangle$ -t. Az y_j méréseket vektorrá fűzve, y , és a ϕ_j^T -ket mint sorok egy $M \times N$ -es Φ mátrixszá fűzve, és ezeket felhasználva (1)-ben, kapjuk, hogy

$$y = \Phi x = \Phi \Psi s = \Theta s, \quad (2)$$

ahol $\Theta := \Phi \Psi$ egy $M \times N$ -es mátrix. A (2)-ben lévő összefüggések a 1.a ábrán figyelhetők meg.[8] Itt érdemes megjegyezni, hogy a mérési eljárás teljesen független, azaz Φ független x -től. A célunk, hogy az általánosított lineáris mérési eljárást használva módszert adjunk arra, hogy egy jel érzékelése és későbbi

rekonstrukciója megtörténhessen. Ehhez egy stabil Φ mérési mátrixra (*measurement matrix*) és egy stabil rekonstrukciós optimalizáló algoritmusra lesz szükség, amelyek segítségével a K -ritka és tömöríthető jelek $M \approx K$ vagy kissé több mérésekből helyreállíthatóak.

A *compressive* adatgyűjtésünk mérési oldalát a Φ mérési mátrix képezi, ezzel kell az \mathbf{x} jelet beszorozni ahhoz, hogy a mérés eredménye, \mathbf{y} , előálljon. Fontos, hogy a Φ mátrix megőrizze a jelentős komponens-információkat a K -ritka jelekben az N -ből M -be történő dimenzióscsökkentés ellenére is. Nyilvánvaló, hogy amennyiben ez nem teljesül, az információ a méréseink során sérül, és az M -hosszú \mathbf{y} -ból nem tudjuk helyreállítani a teljes, N elemből álló \mathbf{x} jelünket, illetve annak ritka Ψ -tartománybeli \mathbf{s} reprezentációját. Φ -t bárhogyan is válasszuk, általában sajnos információvesztés történik. Mivel a mérési eljárásunk lineáris, Φ és Ψ határozzák meg, \mathbf{s} helyreállítása adott \mathbf{y} -ból egy lineáris algebrai feladat, amely $M < N$ esetén, vagyis több ismeretlen, mint egyenlet esetén, nem egyértelműen elvégezhető (létezik zérustér).

Azonban, a jelünk ritkasága mégis segítségünkre lehet. Ebben az esetben ugyanis, \mathbf{y} a Θ csak K oszlopának lineáris kombinációja, amelyekhez tartozó $s_i \neq 0$, lásd 1.b ábra. Ha ismernénk a K nem zérus együttható helyét \mathbf{s} -ben, képesek lennénk formálni egy $M \times K$ méretű egyenletrendszert, amelyet már megfelelően meg tudunk oldani a K darab nem zérus együtthatóra, $M \geq K$. Az egyetlen szükséges és elégséges feltétele a stabil inverz képzésnek és megoldásnak, hogy a Θ mátrix normatartó transzformáció legyen, vagyis bármely K -ritka \mathbf{v} vektorra

$$1 - \epsilon \leq \frac{\|\Theta \mathbf{v}\|_2}{\|\mathbf{v}\|_2} \leq 1 + \epsilon \quad (3)$$

$\epsilon > 0$ esetén. Ez az ún. korlátozott izometria tulajdonság (*restricted isometry property, RIP*).

Természetesen a valóságban sosem ismerjük a K nem nulla együtthatók helyét \mathbf{s} -ben. Mégis, bebizonyították, hogy ha nagyon kicsi ϵ -ra Θ teljesíti a (3)-at minden önkényesen választott $3K$ -ritka \mathbf{v} vektorra, akkor az egyenletrendszereink megoldható a K -ritka \mathbf{s} -re, megfelelően nagy valószínűséggel.

A másik fontos feltétele a stabilitásnak, hogy a mérési mátrix Φ inkoherens legyen a ritka reprezentációt kínáló Ψ mátrixszal, vagyis $\{\phi_j\}$ vektorok ne reprezentálhassák ritkán a $\{\psi_i\}$ vektorokat és fordítva sem. Ismert példa lehet a Dirac-delta impulzus és a Fourier szinuszok ilyen kapcsolata, rendre $\{\phi_j\}$ és $\{\psi_i\}$ szerepekkel. A *Compressive Sensing* témakörében az ún. *mutual coherence* (kölcsonös koherencia) segítségével fejezik ki a két fontos mátrix kapcsolatát. Elvárás, hogy ezen érték minél alacsonyabb legyen, ezzel javítva a rekonstrukció minőségét.

A kérdés tehát, hogy amennyiben ismerünk olyan tartományt, amelyben az érzékelni kívánt jelünk ritka reprezentációval rendelkezik, hogyan adunk meg hozzá olyan Φ mérési mátrixot, amellyel együtt használva a $\Theta = \Phi\Psi$ teljesíti a RIP és inkoherencia tulajdonságokat az adott Ψ mátrixra tekintve. A válasz sajnos az, hogy egyszerűen ez nem tehető meg, hiszen a RIP ellenőrzéséhez mind az $\binom{N}{K}$ darab K -ritka vektort ki kellene próbálni az N -hosszú \mathbf{s} vektorban, amely

kombinatorikusan komplex algoritmust igényel. A CS-ben ezt a problémát Φ véletlen választásával küszöbölik ki. Például, a mátrix $\Phi_{j,i}$ elemeit egy $\mathcal{N}(0, \sqrt{N})$ normális eloszlásból egymástól függetlenül, azonos eloszlással választják. Ekkor az \mathbf{y} mérés lényegében M darab különböző véletlen súlyozású lineáris kombinációja az \mathbf{x} jelnek. A normális eloszlást használva kialakított Φ mérési mátrix nagy valószínűséggel inkoherens a $\Psi = \mathbf{I}$ Dirac-delta bázissal, hiszen mind az N darab impulzusra szükség van ahhoz, hogy egy vektort reprezentáljunk Φ -ben. Továbbá megmutatható, hogy egy $M \times N$ -es, $\Theta = \Phi\mathbf{I}$ véletlen mátrix nagyon nagy valószínűséggel rendelkezik a RIP tulajdonsággal, ha $M \geq cK \log(\frac{N}{K})$, ahol c egy kis értékű konstans. Ezáltal lehetővé válik N -hosszú K -ritka vagy tömöríthető jelek rekonstrukciója mindössze $M \geq cK \log(\frac{N}{K})$ véletlen mérésből, amely látható, hogy jóval kevesebb, mint a Nyquist-törvény által diktált N darab minta. Ezen túl, a Φ véletlen mátrix tulajdonságainak köszönhetően, a $\Theta = \Phi\Psi$ mátrix is véletlen mátrix Ψ választásától függetlenül. Így látható, hogy a véletlen Φ mérési mátrix univerzális abból a szempontból, hogy a $\Theta = \Phi\Psi$ is mindenképpen teljesíti a RIP tulajdonságot a ritkaságot kiváltó Ψ mátrix választásától függetlenül. Az említett véletlen mátrixon túl a véletlen ± 1 értékekkel rendelkező ún. *Rademacher* mátrixok, illetve az $N \times N$ -es Fourier mátrixból véletlenül választott M sorból kialakított mátrixok is rendelkeznek RIP tulajdonsággal. [5, 6]

Amennyiben már rendelkezünk megfelelő M számú és megfelelően elvégzett méréssel, a RIP tulajdonság elméletben garantálja, hogy a K -ritka reprezentációval rendelkező \mathbf{x} vektor információ tartalma megfelelően megőrződött ahhoz, hogy a redukált dimenziójú \mathbf{y} -ból helyreállítható legyen maga \mathbf{x} vagy a vele ekvivalens \mathbf{s} együtthatóvektor. A rekonstrukciós fázis feladata tehát, hogy az eredeti jel legjobb becslését szolgáltatassa, felhasználva ehhez \mathbf{y} -t, Φ -t és Ψ -t.

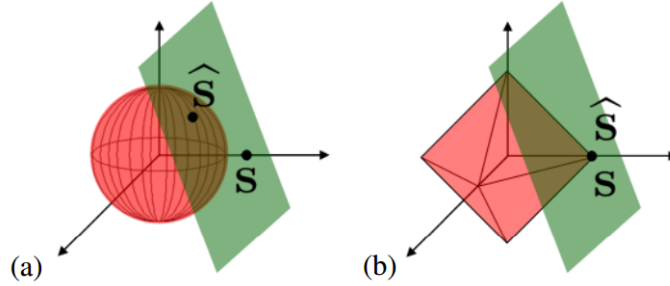
Mivel $M < N$ (2)-ben, ezért végtelenül sok olyan \mathbf{s}' létezik, amely kielégíti a $\Theta\mathbf{s} = \mathbf{y}$ egyenletet. Mindezen \mathbf{s}' egy $N - M$ -dimenziós $\mathcal{H} := \mathcal{N}(\Theta) + \mathbf{s}$ hipersíkon fekszik R^N -ben, amely a Θ mátrix null-terének, $\mathcal{N}(\Theta)$ -nak, felel meg, transzlálva a valódi \mathbf{s} megoldásba. A célunk tehát az, hogy megtaláljuk a ritka, valódi \mathbf{s} együtthatóvektorunkat ebben a hipersíkban.

A továbbiakban definiáljuk az \mathbf{s} vektor ℓ_p normáját, mint $\|\mathbf{s}\|_p = \sqrt[p]{\sum_{i=1}^N |p_i|^p}$. $p = 0$ -ra, az eredmény az ℓ_0 "norma", amely nem valódi vektornorma, a nem zérus elemek számát adja meg. Egy K -ritka vektor ℓ_0 normája tehát K .

A fentiekben tárgyalt egyenletrendszer megoldása a klasszikus módszerrel az ℓ_2 -es norma alapján történik, vagyis a legkisebb-négyzetek módszerével. Ekkor a \mathcal{H} hipersík azon pontját kapjuk meg, amely legközelebb helyezkedik el az origóhoz (2. ábra), vagyis a legkisebb energiával rendelkezik. A megoldandó minimalizálás ℓ_2 -es norma alapján történik:

$$\hat{\mathbf{s}} = \arg \min \|\mathbf{s}'\|_2 \text{ úgy, hogy } \Theta\mathbf{s}' = \mathbf{y} \quad (4)$$

Habár erre a problémára zárt alakban is rendelkezésre áll $\hat{\mathbf{s}} = \Theta^T(\Theta\Theta^T)^{-1}\mathbf{y}$ megoldás, ha mi K -ritka \mathbf{s} megoldásokat keresünk, ezzel a módszerrel szinte sosem a legritkább megoldást találjuk meg. Ezzel szemben ha a ℓ_0 -s normát használjuk, logikusnak tűnik, hogy sokkal könnyebb a ritka megoldás keresése,



2. ábra: A cél a valódi \mathbf{s} , ritka vektor megkeresése. (a) Az ℓ_2 alapú optimalizálás eredménye egy nem ritka $\hat{\mathbf{s}}$ megoldás. (b) Az ℓ_1 alapú minimalizálás eredménye egy ritka $\hat{\mathbf{s}}$ vektor.

hiszen ezzel pontosan a nem nulla elemek száma szerint optimalizálhatunk:

$$\hat{\mathbf{s}} = \arg \min \|\mathbf{s}'\|_0 \text{ úgy, hogy } \mathbf{\Theta s}' = \mathbf{y} \quad (5)$$

Megmutatható, hogy mindössze $M = K + 1$ normális eloszlású egymástól független, azonos eloszlású véletlen méréssel ezen optimalizálási feladat tökéletesen rekonstruálja a K -ritka jelünket nagyon nagy valószínűséggel. Sajnos azonban a megoldása NP-teljes probléma, az összes $\binom{N}{K}$ lehetséges kombináció felsorolását és kipróbálását igényli.

A *Compressive Sensing* térhódításának egyik fő lépése volt, amikor ismertté vált, hogy $M \geq cK \log(\frac{N}{K})$ normális eloszlású egymástól független, azonos eloszlású véletlen méréssel a K -ritka \mathbf{s} jelünk tökéletesen visszaállítható és a tömöríthető jelek jól közelíthetők stabilan, nagyon nagy valószínűséggel ℓ_1 alapú optimalizálással:

$$\hat{\mathbf{s}} = \arg \min \|\mathbf{s}'\|_1 \text{ úgy, hogy } \mathbf{\Theta s}' = \mathbf{y} \quad (6)$$

Ez egy konvex optimalizálási probléma, amely lineáris programozási feladattá alakítható, amely *basis pursuit* néven ismert, számítási komplexitása $\mathcal{O}(n^3)$. Az ℓ_1 norma, az elemek abszolútérték-összege a vektorban, önmagában indukálja a ritkaságot, egy hipersíkon előbb a koordináta tengelyekhez közeli pontokat éri el, ennek szemléltetése látható a 2. ábrán.[8] Összefoglalva tehát elmondható, hogy a *Compressive Sensing* témaköre magába foglalja jelek lineáris mérésének és az ehhez használt $\mathbf{\Phi}$ mátrix megalkotásának elméletét, illetve foglalkozik az inverz feladatok megoldását elősegítő rekonstrukciós optimalizáló eljárások fejlesztésével.

Az egyik első alkalmazása a *Compressive Sensing* elméletnek az ún. egypixel-es kamera (*Single Pixel Camera*. SPC) volt.[9] Ezen rendszer képek rögzítését tette lehetővé CS segítségével. A rögzítendő képet nem egy CCD vagy CMOS lapkára fókuszálja, hanem egy lencsével a képet egy digitális mikrotükrös rendszerre vetíti, amelyben a számos mikrotükrös mindegyike két különböző állapotba állítható. A 1 értékhez tartozó állapotukban a tükrök a beérkező fénysugarakat

egy gyűjtőlencsére továbbítják, még a 0 értékhez tartozó állapotukban nem a gyűjtőlencsébe irányítják. A gyűjtőlencsébe érkező fénysugarakat egy darab fotodiódára fókuszálják, amely egy ADC segítségével méri és digitalizálja a beérkező fény mennyiségét. A tükrörendszert egy véletlen 0/1 számgenerátor hajtja meg, amely a tükröket egyesével a két dedikált állapot valamelyikébe mozgatja. Az 1 állapotban lévő tükrök által visszavert sugarak akkumulált összege jelenik meg a fotodiódán, vagyis fizikailag elvégeztük a jelünk (képünk) és egy véletlen minta belső szorzatának kiszámítását. M különböző mintával meghajtva a tükröt az SPC kimenetén az M darab értékből álló \mathbf{y} vektor jelenik meg. A feladat a véletlen minták és \mathbf{y} ismeretében az eredeti kép helyreállítása. Feltételezhető, hogy az eredeti képnek van ritka reprezentációja valamely ismert (pl: wavelet) tartományban. Az ily módon megalkotott rendszer jóval a Nyquist-ráta alatti, M darab méréssel képes volt helyreállítani N pixelből álló képeket, $M \ll N$. Nagy előnye továbbá a rendszernek, hogy a spektrum olyan tartományaiban is képes képeket készíteni, amelyekben hagyományos, pixeles kamerával nagyon drága lenne a felvétel készítés, ugyanis itt csak egyetlen pixel megalkotására van szükség. Infravörös és gamma tartományban egy fényérzékelő elem előállítási költsége sokkal magasabb, mint látható fény esetében.

A CS az elmúlt tíz évben a tudomány számos területén megjelent. Radar-berendezések jelfeldolgozását forradalmasította, ahol a hihetetlenül nagy számú minta gyors rögzítése (akár 5 milliárd minta/másodperc) már a technológia korlátait súrolta.[10] CT és MRI berendezésekben, ahol a minták készítése "drága", CS segítségével, felbontásbeli romlás nélkül a vizsgálatok kevesebb mérésből és idő alatt elvégezhetőek.[11, 12] Jelentős szerepet kapott a CS a vezeték nélküli szenzorhálózatokban, ahol események érzékelését jóval gazdaságosabban meg lehetett oldani a segítségével, a mintavételi ráta akár a 80%-kal is csökkenthető, amely mind energia, mind sávszélesség kihasználás szempontjából sokkal előnyösebb.[13–16] A CS megtöri a mára berögzült konvenciót, miszerint az érzékelés helyén tömörítünk, a helyreállítás pedig már egyszerű és gyors. Az új érzékelési paradigma értelmében a jelek mintavételezése egyből tömörítést is jelentene, vagyis a szenzorok méginkább egyszerűsödhetnek és a számítási feladatok az erőforrásokban bővelkedő bázisállomásokon, adatközpontokban összpontosulnának az erőforrásszegény eszközök helyett. Hátrányként szokás említeni, hogy a CS valós időben nem képes jelek érzékelésére, mert a teljes \mathbf{x} jelet előbb mérni kell ahhoz, hogy azon utána a lineáris méréseket el tudjuk végezni. Módszerek már léteznek, amelyek feloldják az említett korlátozást és CS elven működő ADC berendezések publikálása megtörtént. Ezekre általában *analog-to-information converter*-ként, vagyis analógból információvá alakítókként hivatkoznak.[17]

A képfeldolgozás számos területén is alkalmazzák a CS technikáit, főleg az ℓ_1 -en alapuló rekonstrukcióhoz kapcsolódó eredményeket.[18, 19] Ezek közé tartozik például a képek zajmentesítése (*image denoising*), amely során a zajjal terhelt képek helyreállítása a cél. A másik hasonló feladat a képek kiegészítése, hiányzó képrészletek helyes kitöltése (*image inpainting*). A képek nagyítása (*image up-scaling*) is a vizsgált területek közé tartozik.

A CS a vizsgált területek mindegyikén *state of the art* eredményként szerepel. Térhódításának mégis számos tényező akadályt szab. Az egyik a nehézkes használata és a bonyolult matematikai háttere. A másik a tesztelhetőségének nehézsége. Mivel a mai hardverek mind a hagyományos érzékelési paradigma szerint működnek, nagyon nehéz a CS teljesítményét valós körülmények közt tesztelni, ezért sok CS-gel foglalkozó kutató kénytelen a fejlesztett módszereket más célra rögzített adatbázisokon kiértékelni. A másik lehetséges választás valamely egyedileg, erre a célra tervezett vagy már meglévő rendszer nehézkes ilyen célú kialakítása, átalakítása. Ehhez azonban beágyazott rendszerekkel kapcsolatos ismeretekre van szükség, amely sokszor nem képezi valódi tárgyát a CS témakörébe tartozó kutatásoknak.

3. Probléma

A kutatásom célja egy olyan általános hardver- és szoftverrendszer tervezése, amely megfelelő támogatást adhatna a *Compressive Sensing* eredményeinek valós körülmények közötti egyszerűbb teszteléséhez, különös tekintetben a vezeték nélküli szenzorhálózatokkal kapcsolatos problémák megoldása során, erre szolgáló rendszer ugyanis jelenleg nem létezik.

A megfelelő felépítésű és rugalmas kialakítású, alacsony költségű és széles körben elérhető hardver megtervezése, kiválasztása fontos lépés a rendszer használhatóságát és elterjedését tekintve. A hardverhez kapcsolódóan a megfelelően strukturált és kialakított szoftver megtervezése is elengedhetetlen. Mind az érzékelési, mind a rekonstrukciós oldal szoftvere gondosan megtervezésre kell, hogy kerüljön a későbbi minél rugalmasabb használhatóság érdekében. A következő fejezetekben a hardverhez és szoftverhez kapcsolódó megfontolások tárgyalása található.

4. Hardverrel kapcsolatos tervek

A *Compressive Sensing* lehetővé teszi az eddigieknél egyszerűbb szenzorok megépítését. Azáltal, hogy erős asszimetriát tesz lehetővé a számítási kapacitások tekintetében, vagyis a nagy erőforrásigényű feladatokat az érzékelő oldalról a rekonstrukciós oldalra migrálja, újabb előrelépés következhet be a vezeték nélküli szenzorhálózatok világában. A vezeték nélküli szenzorhálózatok (*wireless sensor networks*, *WSN*) ún. *mote*-okból, vagyis kis, egyszerű számítógépekből állnak, amelyek rádióüzenetek segítségével kommunikálnak egymással. Jellemzőik ezen eszközöknek, hogy nagyon alacsony energiaigényűek, a számítási kapacitásuk és a rendelkezésre álló memória mérete is korlátos. Általában olyan elosztott érzékelési feladatok megoldására használatosak, amelyekben a hosszú élettartam és megbízhatóság elsődleges szempont. Napjainkban a szenzorok mind a hagyományos jelfeldolgozási paradigma szerint érzékelik környezetüket, amelyhez olykor nagyobb memóriára és a tömörítés gyors végrehajtásához nagyobb fogyasztású CPU-ra van szükség. A fogyasztás növelése mellett az eszközök árát is nagyban befolyásolja az említett két paraméter. A CS alkalmazásával olyan

tömörítő jelérzékelésen alapuló szenzorok használata válna lehetővé, amelyek az érzékelés során a tömörítést is elvégeznék minden egyéb algoritmus futtatása nélkül.

Az említett előnyök miatt a CS-gel kapcsolatos kutatások egyik fő célpontja lett a WSN-ekben való alkalmazás lehetősége. Rengeteg megoldás született, amelyek mind a WSN-ekben felmerülő tipikus feladatokra az eddieknél jóval jobb eredményt adtak.[13–16] Ezek kiértékelése általában speciálisan átalakított eszközökön történt, illetve szimulációk, meglévő adatbázisok felhasználásával. A CS egyik fő előnyeként azt szokás említeni, hogy jóval optimistább, mint a hagyományos, Nyquist-alapú jelérzékelés, vagyis a természet általában megismerhető ritka jelek segítségével, és az információs ráta alacsonyabb, mint a jelváltozási ráta. Gyakorlati alkalmazásokban a CS jobb eredményt ad, mint az elméletben bebizonyított garanciák. Emiatt nagyon fontos lenne, hogy kutatók, a kifejlesztett módszereiket valós körülmények közt, egyszerűen tesztelni tudják.

Napjainkban még nem állnak rendelkezésre CS paradigmán alapú hardver egységek, ezért ezek működésének emulálásával történnek a kutatások. Az általam tervezett hardver a WSN-ekben felmerülő feladatok CS-gel történő megoldásának teszteléséhez, kiértékeléséhez nyújtana segítséget. A valódi felhasználhatóság érdekében a tervezett eszköznek teljesítenie kell a WSN-ekben előforduló mote-okkal szemben támasztott követelményeket. Fontos, hogy olcsó és elterjedt architektúráról legyen szó, amely megfelelő CPU-ra épül. WSN-ekben a CPU szerepét egy nagyon korlátozott erőforrásokkal rendelkező, 8-bites mikrovezérlő szokta betölteni. Esetünkben azonban a CS hardverek emulálásához olykor nagyobb erőforrásra lehet szükség (pl. képekkel történő munka során). A megfelelő CPU számunkra tehát egy közepes teljesítményű mikrovezérlő, amely megfelelően nagy memóriával rendelkezik, számítási kapacitása a fogyasztásához képest kiemelkedően magas és rendelkezik olyan mélyalvó állapotokkal, amelyekben a fogyasztása drasztikusan csökkenthető. Előnyt jelent a számos általános célú ki-bemeneti vonal megléte és sok ADC használhatósága. Maga a kialakítandó eszköz akkor lehetne a legelterjedtebb és sokak számára egyszerűen használható, ha már napjainkban is populáris rendszerről lenne szó, amely már létező felhasználói körrel, implementált funkciókkal rendelkezik. Elterjedtségének következménye, hogy már számos kiegészítő modul, érzékelő, beavatkozó elérhető, amelyek kezelése függvénykönyvtárak segítségével megoldott, ami elősegíti a nem szakemberek számára is az egyszerű használatukat.

A kialakítandó hardver tehát egy, már létező elemekből (*off-the-shelf*) megépülő rendszer, amely jól dokumentált, tervei elérhetőek (*open-hardware*), hogy amennyiben valaki átalakítani, saját céljaira szabni szeretné, azt könnyedén megtehesse. Emellett az ára is kiemelkedően fontos. WSN-ekben a kommunikáló eszközök száma sokszor magas is lehet (akár százas nagyságrend). Több eszköz egyidejű használata a kifejlesztett módszerek relevánsabb tesztelését tenné lehetővé. A széleskörűen megvásárolható kommunikációs modulok (*WiFi*, *ZigBee*, *BlueTooth*) és azok egyszerű használata, hálózatba szervezése is elsődleges szempont. A kommunikáció szávszélessége WSN-ekben erősen korlátozott, emiatt is

van szükség tömörítésre, illetve energiaigénye is magas. A cél mindig a lehető legkevesebb üzenettel és adattal egy probléma minél robosztusabb megoldása.

A felsorolt követelmények nagy részét sok, napjainkban populáris rendszer támogatja. Közülük is talán kiemelkedik az ún. *Arduino* rendszer.[20] Magában foglalja a hardver és szoftver rendszer felépítését és kezelését is. Számos bővítő modullal rendelkezik, amelyek működtetéséért felelős könyvtárak jórészt rendelkezésre állnak. Több fajta CPU absztrakt használata lehetséges, tehát nem szükségszerű beágyazott rendszerekkel kapcsolatos ismeretek megléte. *Open-hardware* lévén bárki átalakíthatja saját igényeire, ha speciális funkcióra van szükség. A felhasználói közössége kiemelkedően nagy létszámú, ára rendkívül kedvező. Megfelelő szoftver támogatással a tervezett *Compressive Sensing* tesztkörnyezet hardver alapjául szolgálhatna. Másik lehetőség a már ismert vezeték nélküli szenzorok használata, mint pl. a TelosB, MicaZ, Iris.[21] Ezek azonban nagyon alacsony számítási kapacitással rendelkeznek ahhoz, hogy CS szenzorok működését elegendően gyorsan emulálni tudják. Az általam választott platform az Arduino család *Due* nevezetű tagja, amely egy 84MHz-es órajellel működő Atmel SAM3X8E ARM Cortex-M3 CPU-t tartalmaz, amelyet alacsony fogyasztás és gyors megszakításkezelés jellemez.[22]

5. Szoftverrel kapcsolatos tervek

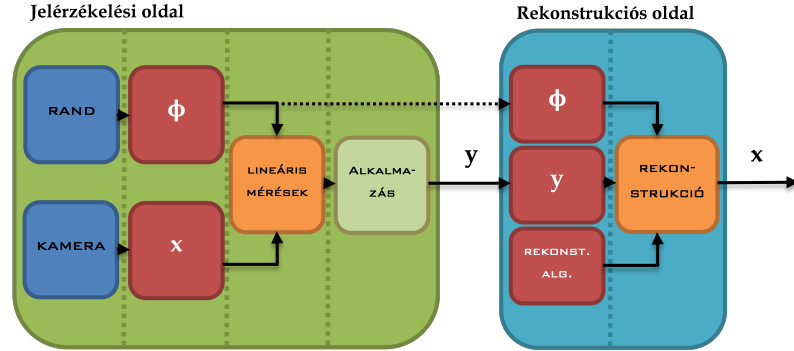
A hardver kapcsán említett előnyök jó kihasználhatósága érdekében a szoftver rendszernek kezelnie kell a nagyfokú rugalmasságot, ugyanakkor biztosítani kell, hogy az alacsony szintű, hardver-közei funkciók a rendszer felhasználója számára rejtve legyenek. Ezen alapvető funkciók könnyű implementálhatóságát és jó egységbe zárását azonban biztosítani kell. Ennek megfelelően a kialakítandó rendszer moduláris szerkezettel kell, hogy rendelkezzen, amelyen belül az egyes modulok jól elhatárolt feladatokat valósítanak meg és csak interfészek vagy üzenetek segítségével kérhetők fel ezen feladatok elvégzésére. Mivel a vezeték nélküli szenzorok sokszor időkritikus feladatokat végeznek, ezért valós idejű operációs rendszer használata a problémák megoldásában sokszor elengedhetetlen. A legmegfelelőbb szoftver egy olyan valós idejű operációs rendszer lenne, amely önmagában képes a CS-gel kapcsolatos feladatok ellátására és támogatást nyújtana a lineáris mérések elvégzésében. A kutatók feladata ekkor csak egy magas szintű alkalmazás fejlesztése lenne, amely a megfelelő modulokat használva elvégezne összetettebb feladatokat. Az alacsony szintű modulok támogatnák a CS elvégzéséhez szükséges funkciók végrehajtását, vagyis a Φ mérési mátrix megalkotását, az adatok beolvasását, a lineáris mérések elvégzését és az eredmény vektor előállítását. Minden egyes modulból néhány ismertebb megoldás előre implementálásra kerülne, de dokumentációs segítséggel a kutatók új, hasonló feladatot elvégző modulokat is létrehozhatnak. Például a véletlen számok generálásáért felelős modul is több különböző elven működhetne, és mindenki a saját alkalmazásához megfelelőt választhatja, vagy implementálhatja saját megoldását. Egy CS alkalmazás tehát csak modulok huzalozásából állna, nem foglalkozna a mögöttes logikával, a fejlesztő megadhatja az adat forrását képviselő modult

(pl. ADC), a használandó Φ mérési mátrix generálását végző modult és ezeket egy általános lineáris méréseket végző modulba köti, amelynek eredménye az M darab értékből álló mért értékek \mathbf{y} vektora.

Az eddigiekben ismertetett struktúra több valós idejű operációs rendszerben is kialakítható. Egyik ismert példa moduláris felépítésű valós idejű operációs rendszerre a *TinyOS*. [23] A TinyOS egy eseményvezérelt valós idejű operációs rendszer, amely *NesC* nyelven íródott. A NesC a C nyelvhez hasonló szintaxissal rendelkezik, de a fejlesztést moduláris alapokra helyezi. Két alapvető egysége van, a *modul* és *konfiguráció*. A modulok funkciókat valósítanak meg, a konfigurációk a modulok összehuzalozását végzik. A kommunikáció modulok közt előre definiált interfészekon keresztül történik. A TinyOS kis méretű, egyszerű logikát követő operációs rendszer, amely WSN eszközök számára lett kifejlesztve. Kiegészítve a CS-hez kapcsolódó elemekkel megfelelő megoldást nyújthatna a tesztkörnyezet mérési oldalán lévő szoftver problémájára. Egyéb lehetőségként használható lenne bármely valós idejű, beágyazott operációs rendszer, mint pl. a Contiki [24], FreeRTOS [25], RIOT [26]. Ezekben azonban az említett moduláris szerkezet kialakítása sokkal nehezebb. A tényleges szoftver kialakítását TinyOS-ben kezdtem meg.

A szoftver rendszer a CS témakörében azonban nem csak az érzékelő oldalon érdekes, hanem a rekonstrukció során is fontos szerepet kap. A jelek helyreállításáért felelős algoritmusok nagyobb számításigényűek, tehát ezeket PC-n vagy szerverszámítógépen érdemes futtatni. Mivel már sok ezek közül több nyelven is implementálásra került, ezért ezek összegyűjtése és megfelelő használata az elsődleges elvárás a rekonstrukciós szoftvertől. Egy olyan absztrakt alkalmazás megírása a cél, amelynek definiálni tudunk egy adatforrást (elvégzett mérések), megadhatjuk a lineáris mérések során használt mérési mátrix generálásának elvét és választhatunk a rekonstrukciós algoritmusok széles tárházából. A szoftver elvégzi az adatok megfelelő formába történő átalakítását, meghívja a megadott rekonstrukciós eljárást és kimenetén megjelenik a helyreállított jel. Külön lehetőségként elképzelhető, hogy az alkalmazás felhő alapú szolgáltatásként fusson, amely lehetővé tenné adatok egyszerű összegyűjtését és több adatforrás párhuzamos feldolgozását.

A szoftverrendszer felépítését az 3. ábrán figyelhetjük meg, egy kamera képezne CS alapú rekonstrukciójának menetét követhetjük végig. A két fő komponenst a mérési és rekonstrukciós oldal alkotja. A mérést végző eszköz szoftvere az elmondottak alapján rétegzett felépítésű, melyek határait a szaggatott függőleges vonalak jelzik. Itt a legalsó rétegben a kamera kezelését és a kép átvételét végrehajtó, illetve a véletlen számok generálását végző komponensek kapnak helyet. A következő réteg moduljai már ezeket használva megalkotják az \mathbf{x} mérendő vektort és a Φ mérési mátrixot. Ezeket felhasználva a felsőbb réteg elvégzi a lineáris méréseket, amelyeket majd az alkalmazás rétege fog feldolgozni és továbbítani megfelelő kommunikáció segítségével az adatgyűjtő állomásra. A rekonstrukciós oldal megkapja a mért \mathbf{y} vektort, illetve a méréshez használt Φ mátrixot, vagy annak generálásához szükséges adatokat. A megfelelő rekonstrukciós algoritmus kiválasztása után a jel helyreállítása következik.



3. ábra: Tervezett szoftverfelépítés: bal oldalon a mérési oldal, amely rétegzett felépítésű és CS elven érzékeli x -et, jobb oldalon a rekonstrukciós oldal, amely a CS mérésekből becsüli az eredeti x -et.

6. Összefoglalás

A *Compressive Sensing* paradigmaváltást hozhat a jelérzékelés világába. A Nyquist-ráta szerint gyűjtött minták számától jóval kevesebb mérésekből képesek vagyunk helyreállítani egy jelet, kihasználva annak azon tulajdonságát, hogy ritka reprezentációval rendelkezik valamely ismert tartományban. A CS asszimmetriát teremt a szenzoron és a bázisállomáson elvégzendő számítások tekintetében, a nagy erőforrásigényű feladatokat a nagyobb kapacitású eszközökre migrálva. Jelenleg rengeteg kutatás foglalkozik a CS alkalmazásaival, amelyeknek kissé határt szab a CS alapú hardverek, szenzorok hiánya. Emiatt sok kutatás nem valós körülmények közt teszteli a kifejlesztett módszereket. Célom egy olyan tesztberendezés és hozzá tartozó szoftver terveinek lefektetése volt, amelyeket követve egy általánosan használható tesztkörnyezet megépíthető és használható CS algoritmusok, eredmények valódi kiértékelésére. Jelen munka a *Compressive Sensing* bemutatásával és a tervek ismertetésével foglalkozik.

Irodalom

1. Lawrence R Rabiner and Bernard Gold. Theory and application of digital signal processing. *Englewood Cliffs, NJ, Prentice-Hall, Inc., 1975. 777 p., 1, 1975.*
2. E. J. Candes, J. Romberg, and T. Tao. Robust uncertainty principles: exact signal reconstruction from highly incomplete frequency information. *IEEE Transactions on Information Theory*, 52(2):489–509, Feb 2006.
3. D. L. Donoho. Compressed sensing. *IEEE Transactions on Information Theory*, 52(4):1289–1306, April 2006.
4. Emmanuel J. Candès. The restricted isometry property and its implications for compressed sensing. *Comptes Rendus Mathématique*, 346(9):589 – 592, 2008.
5. Yonina C Eldar and Gitta Kutyniok. *Compressed sensing: theory and applications*. Cambridge University Press, 2012.

6. Simon Foucart and Holger Rauhut. *A mathematical introduction to compressive sensing*, volume 1. Springer, 2013.
7. E. J. Candes and M. B. Wakin. An introduction to compressive sampling. *IEEE Signal Processing Magazine*, 25(2):21–30, March 2008.
8. Richard G Baraniuk. Compressive sensing. *IEEE signal processing magazine*, 24(4), 2007.
9. Marco F Duarte, Mark A Davenport, Dharmpal Takhar, Jason N Laska, Ting Sun, Kevin E Kelly, Richard G Baraniuk, et al. Single-pixel imaging via compressive sampling. *IEEE Signal Processing Magazine*, 25(2):83, 2008.
10. R. Baraniuk and P. Steeghs. Compressive radar imaging. In *2007 IEEE Radar Conference*, pages 128–133, April 2007.
11. Guang-Hong Chen, Jie Tang, and Shuai Leng. Prior image constrained compressed sensing (piccs): a method to accurately reconstruct dynamic ct images from highly undersampled projection data sets. *Medical physics*, 35(2):660–663, 2008.
12. M. Lustig, D. L. Donoho, J. M. Santos, and J. M. Pauly. Compressed sensing mri. *IEEE Signal Processing Magazine*, 25(2):72–82, March 2008.
13. S. Li, L. D. Xu, and X. Wang. Compressed sensing signal and data acquisition in wireless sensor networks and internet of things. *IEEE Transactions on Industrial Informatics*, 9(4):2177–2186, Nov 2013.
14. Jia Meng, Husheng Li, and Zhu Han. Sparse event detection in wireless sensor networks using compressive sensing. In *Information Sciences and Systems, 2009. CISS 2009. 43rd Annual Conference on*, pages 181–185. IEEE, 2009.
15. Jin Wang, Shaojie Tang, Baocai Yin, and Xiang-Yang Li. Data gathering in wireless sensor networks through intelligent compressive sensing. In *INFOCOM, 2012 Proceedings IEEE*, pages 603–611. IEEE, 2012.
16. Waheed Bajwa, Jarvis Haupt, Akbar Sayeed, and Robert Nowak. Compressive wireless sensing. In *Proceedings of the 5th international conference on Information processing in sensor networks*, pages 134–142. ACM, 2006.
17. S. Kirolos, J. Laska, M. Wakin, M. Duarte, D. Baron, T. Ragheb, Y. Massoud, and R. Baraniuk. Analog-to-information conversion via random demodulation. In *2006 IEEE Dallas/CAS Workshop on Design, Applications, Integration and Software*, pages 71–74, Oct 2006.
18. Julien Mairal, Francis Bach, and Jean Ponce. Sparse modeling for image and vision processing. *arXiv preprint arXiv:1411.3230*, 2014.
19. K. Egiazarian, A. Foi, and V. Katkovnik. Compressed sensing image reconstruction via recursive spatially adaptive filtering. In *2007 IEEE International Conference on Image Processing*, volume 1, pages I – 549–I – 552, Sept 2007.
20. Arduino. <https://www.arduino.cc/>. Megtekintve: 2016-10-15.
21. Ram Prasad Narayanan, Thazath Veedu Sarath, and Vellora Veetil Vineeth. Survey on motes used in wireless sensor networks: Performance & parametric analysis. *Wireless Sensor Network*, 8(04):67, 2016.
22. Arduino Due, Atmel SAM3X8E ARM Cortex-M3. www.atmel.com/devices/AT-sam3x8e.aspx. Megtekintve: 2017-01-08.
23. TinyOS operációs rendszer. <https://en.wikipedia.org/wiki/TinyOS>. Megtekintve: 2016-10-15.
24. Contiki operációs rendszer. www.contiki-os.org/. Megtekintve: 2017-01-08.
25. FreeRTOS operációs rendszer. www.freertos.org/. Megtekintve: 2017-01-08.
26. RIOT operációs rendszer. <http://www.riot-os.org/>. Megtekintve: 2017-01-08.